Some Things You Always Needed to Know About Systems Engineering but Didn't Know You Needed to Know

PM Challenge Conference

February 2008

M.G. Ryschkewitsch

NASA Chief Engineer

There is no cookbook which can guarantee success but there are necessary ingredients

Completeness

Information, tools, techniques

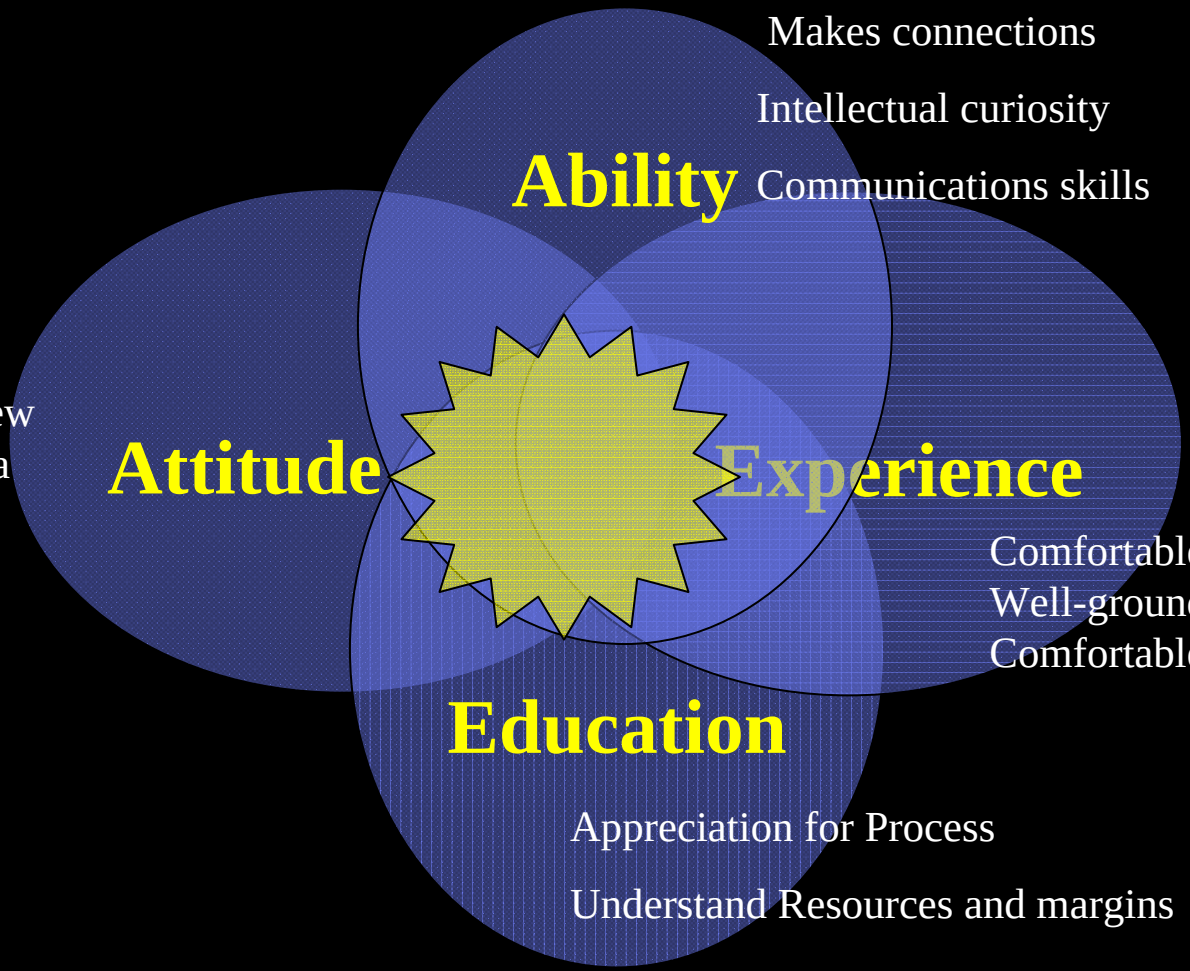
Consistency

Policy, Process

Creativity

Competencies
Human Behaviors
Culture of rigorous inquiry

A Systems Engineer requires a variety of capabilities- some can be learned in class, others are personal and must be developed



Makes connections

Intellectual curiosity

Communications skills

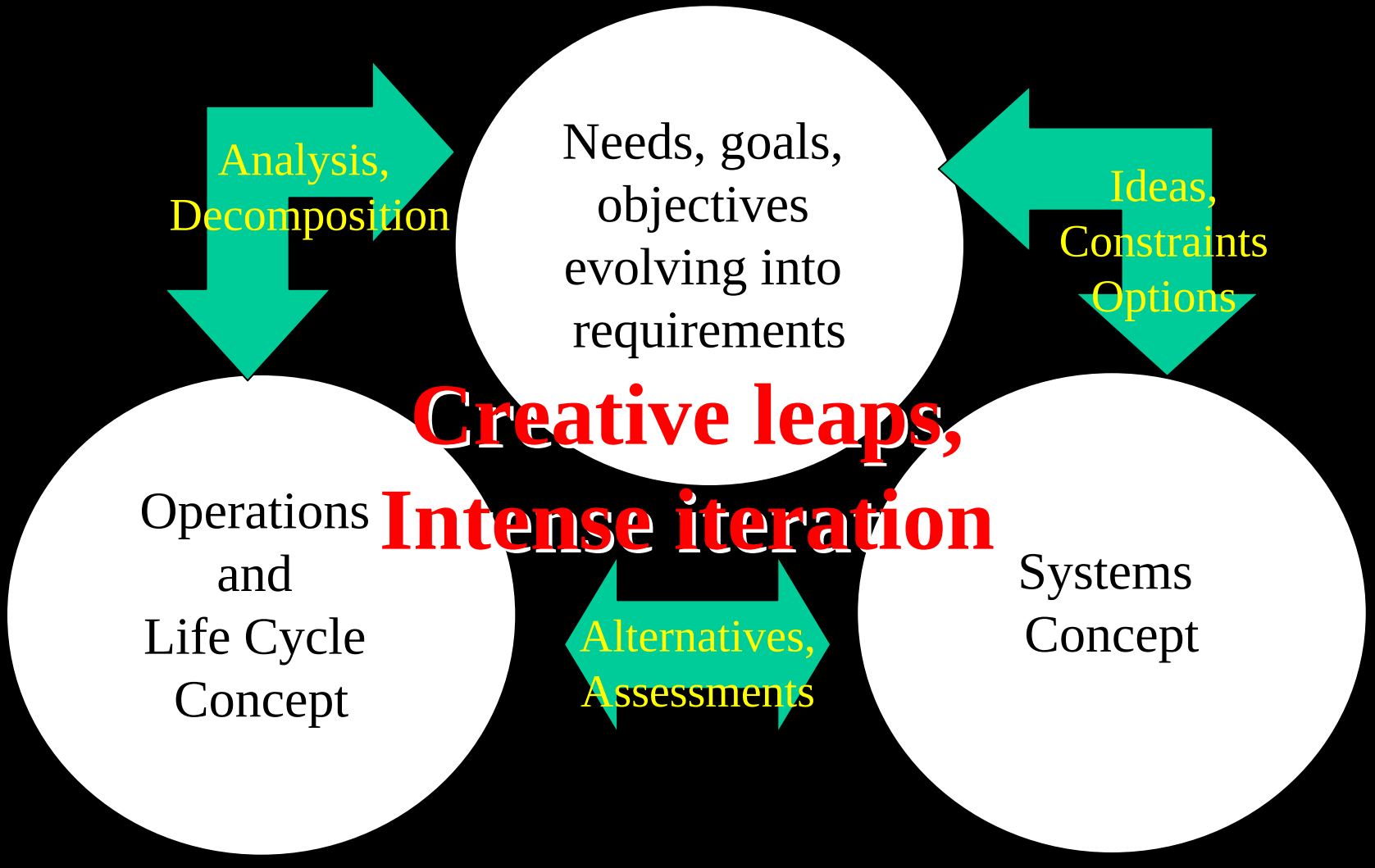
Big Picture View
Proper paranoia
Self confident

Comfortable with Change
Well-grounded in relevant work
Comfortable with uncertainty

Appreciation for Process

Understand Resources and margins

Creation of new systems requires engineering science, art and lots of perspiration



All requirements sets are wrong, some are useful
(with apologies to George Box)

- Working to requirements is essential for good engineering but they must continually be validated
  - If we ever stop asking “Does this makes sense” we are poised for failure
- There are no such things as concept or operations independent requirements; One must always play the requirements and evolving design against the operations planning
- English is a lousy engineering language
- Analyses and models are always approximations

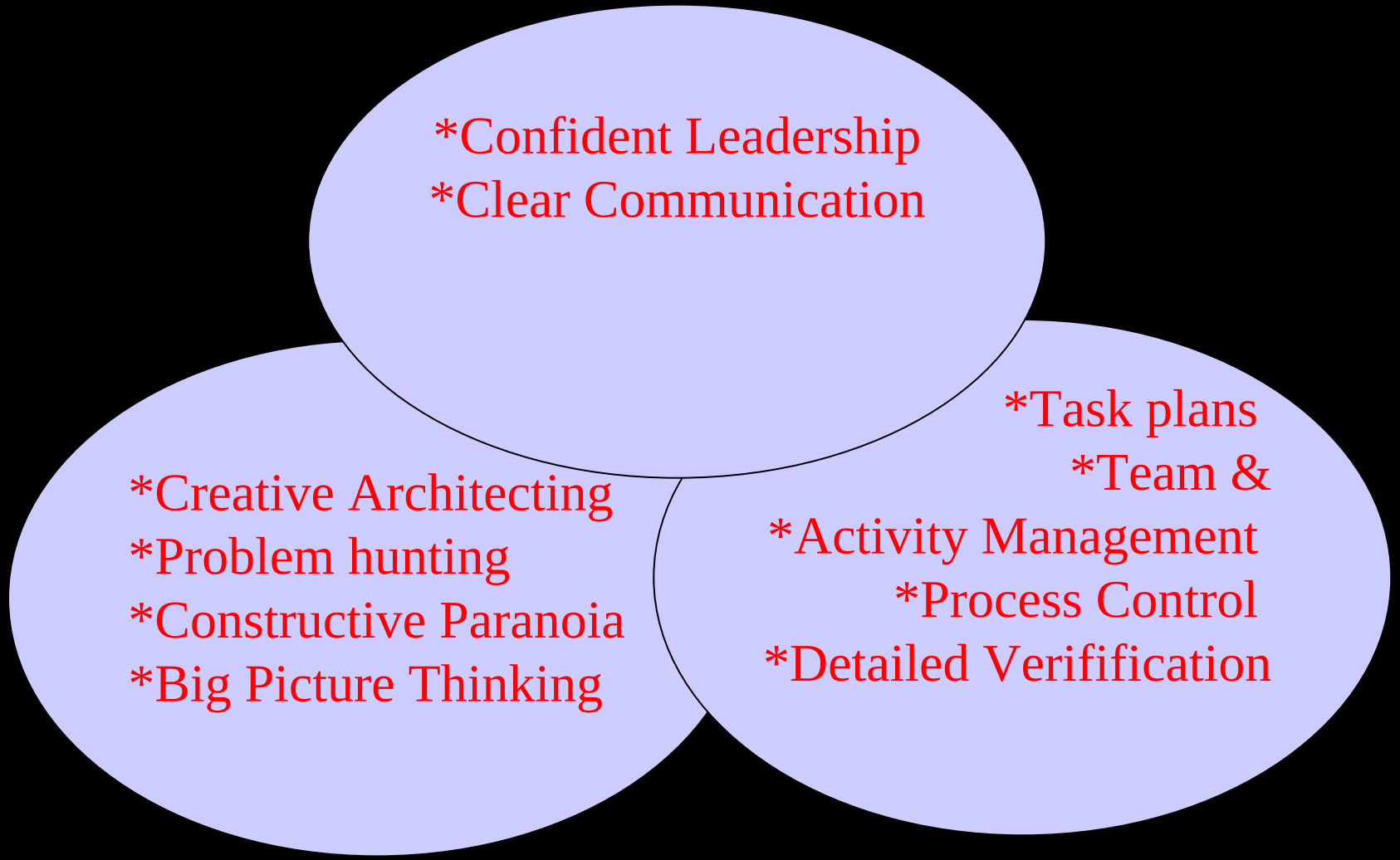
“Test as you will fly” helps but it is just as important to know what the tests don’t tell you!

”Success is not simply the absence of failure, it also masks potential modes of failure” -- Henry Petroski

* Complex systems can fail catastrophically in ways which are impossible to predict
* We must allow for both the known unknown and the unknown unknown
* In the space business, no two systems can be the same; heritage is imperfect at best and usually less than we think
* What is the simplest possible “fail to” configuration which allows us to survive and fight another day
* How good are our tests and what do they NOT tell us
* How much capability do we have beyond the requirement and how can we use that when things change

We must design for success by designing for faults and failures

Complete Systems Engineering is Key to Success



Suggested reading

----Mike Griffin, speech on the two cultures of engineering at Purdue University, March 28, 2007
http://www.nasa.gov/news/speeches/admin/mg_speech_collection_archive_2.html

----Ferguson, E.S. (1992) Engineering in the Mind's Eye. Cambridge, MA: MIT Press.

----Henry Petroski

- "Success through Failure: The Paradox of Design", Princeton University Press, 2006
- "To Engineer is Human", Vintage Books, 1992